The Chinese University of Hong Kong
Department of Computer Science and Engineering

Ph.D. – Term Paper

Title:	Nonrigid Surface Modelling and Fast Recovery
Name:	Jianke Zhu
Student I.D.:	06239910
Contact Tel. No.:	3163-4257
Email A/C:	jkzhu@cse.cuhk.edu.hk
Supervisor:	Prof. Michael R. Lyu
Markers:	Prof. A & Prof. B
Mode of Study:	Full-time
Submission Date:	April 20, 2007
Term:	2
Fields:	
Presentation Date:	TBD
Time:	(TBD)
Venue:	TBD, Ho Sin-Hang Engineering Building

Nonrigid Surface Modelling and Fast Recovery

Abstract

Modelling and recovering nonrigid surface are interesting research problem for image analysis and computer vision, which can be applied in a variety of applications. In this report, two different approaches to nonrigid surface recovery have been proposed. One is based on the appearance models, the other employs the local features.

As for the appearance-based method, a two-stage scheme method is presented for online nonrigid surface recovery toward Augmented Reality (AR) applications using Active Appearance Models (AAMs). First, 3D shape models are constructed from AAMs offline, which do not involve processing of the 3D scan data. Based on the computed 3D shape models, an efficient online algorithm is proposed to estimate both 3D pose and non-rigid shape parameters via local bundle adjustment for building up point correspondences. This approach, without manual intervention, can recover the 3D non-rigid shape effectively from either real-time video sequences or single image. The recovered 3D pose parameters can be used for AR registrations. Furthermore, the facial feature can be tracked simultaneously, which is critical for many face related applications. Promising experimental results demonstrate the proposed scheme is effective for real-time AR applications.

For the feature-based approach, a progressive finite Newton optimization scheme is proposed for the nonrigid surface detection problem, which is reduced to only solving a set of linear equations. The key of this method is to formulate the nonrigid surface detection as an unconstrained quadratic optimization problem which has a closed-form solution for a given set of observations. Moreover, a progressive active-set selection scheme is employed, which takes advantage of the rank information of detected correspondences. Extensive experiments have been conducted for performance evaluation on various environments, whose promising results show that the proposed algorithm is more efficient and effective than the existing iterative methods.

Contents

List of Figures

.

List of Tables

.

Chapter 1

Introduction

1.1 Introduction

Nonrigid surface modelling and detection are essentially the computer vision tasks in a variety of applications in image alignment, medial imaging, augmented reality, human computer interaction and digital entertainment. Extensive research efforts in the computer vision domain have focused on the problems related to deformable object modeling and tracking [2, 7, 34, 42, 43, 45, 46]. Such interests in deformable object tracking are closely related to problems such as image registration, feature matching and object recognition.

For the nonrigid surface, the regularized deformable models are vitally important for dealing with the problem of the many outliers and ill-posed optimization, and thus are able to make the problem tractable by constraining the search space. Various regularization methods have been proposed in the literature, such as the Thin-Plate Spline, the Finite Element Model (FEM) and the data embedding method. The Thin-Plate Spline is well-known interpolation method widely used in point set registration, and mainly penalizes the second order derivatives. Alternatively, the FEM based regularization approach has been extensively studied in [13, 34, 18, 46], which reported promising results in fitting noisy image data and handling deformable 3D objects. Using the FEM models, the nonrigid surface should be explicitly represented by a triangulated mesh. Finally, the data embedding techniques, such as Principal Component Analysis (PCA) [41, 42], can also be engaged as the regularization technique.

As for a successful nonrigid surface detection algorithm, it can be applied to many applications. For example, Active Appearance Models was employed as the face alignment tool in face annotation and face recognition [44], and re-texture a person's face, as shown in Fig. 1.1. Moreover, the nonrigid surface registration can be applied to some more generic and deformable objects, as shown in Fig. 1.2.

(a) Face alignment [44] (b) Re-texturing a face [42]

Figure 1.1. Applications of the Active Appearance Models

(a) T-shirt (b) Paper

Figure 1.2. Detecting nonrigid surfaces in real-time video [46]. (b) T-shirt. (d) A piece of paper.

1.2 Contributions

This report intends to develop the techniques that can effectively and efficiently represent and recovery the nonrigid surface. To this end, two different approaches have been proposed. The main contributions of this report can be further described as follows:

- **Appearance-based method.** In [42], an efficient online algorithm is proposed to estimate both 3D pose and non-rigid shape parameters via local bundle adjustment for building up point correspondences. The proposed approach, without manual intervention, can recover the 3D non-rigid shape effectively from either real-time video sequences or single image. The recovered 3D pose parameters can be used for Augmented Reality registrations. Furthermore, the facial feature can be tracked simultaneously, which is critical for many face related applications, such as automated face modelling [45].

- **Feature-based method.** In [46], a novel progressive finite Newton approach is proposed to tackle the nonrigid surface detection, which is based on the feature correspondences. Moreover, a 2D deformable mesh model is used to constrain the surface deformation. This method has been evaluated on various applications, such as real-time re-texturing the nonrigid surface and medical image registration.

1.3 Organization

This report presents two different approaches to the nonrigid surface modelling and fast recovery. One is an appearance-based method, the other is based on the feature correspondences. The rest of this report is organized as follows. Chapter 2 presents a novel two-stage scheme for online nonrigid surface recovery through fitting Active Appearance Models (AAMs). This method has been applied to Augmented Reality applications and re-texturing a human face. Chapter 3 propose a progressive finite Newton optimization scheme for automatically recovering the nonrigid surface. In contrast to the state-of-the-art method [34], the proposed method more efficient. Moreover, it easy to be implemented, and does not require tuning the viscosity parameters. Note that each chapter of this report is intended to be self-contained.

Chapter 2

Two-stage Scheme with Active Appearance Models

In this chapter, we propose a novel two-stage scheme for online nonrigid surface recovery toward AR applications using Active Appearance Models (AAMs).

2.1 Previous Work and Motivation

The objective of Augmented Reality (AR) is to integrate virtual objects into real-world video sequences, enabling computer generated objects to be overlaid on the video in such a manner as to appear part of the viewed 3D scene. Recently, some well-known AR toolkits have been developed for AR applications [21]. Although these tools have facilitated the AR applications to obtain good registration data automatically and robustly, it is still a challenging and open issue to keep track of objects with their movement, orientation, size, and position accurately in AR applications. This critical requirement also results in an important problem, i.e., determining the position and orientation of an object, which plays an important role in many research areas such as robotics, computer vision, computer graphics.

In the subsequent part we describe some recent advances of technologies for object tracking and shape recovery in the computer vision community. Along with the introduction of previous work, we provide the motivation and brief introduction of our work in this chapter particularly for AR applications.

L. Vacchetti et al. [38] proposed an efficient real-time solution for tracking rigid objects in 3D scene using a single camera. They demonstrated that the virtual glasses and masks can be added on to the head. Since they employed a rigid 3D model, the local facial feature was not able to be located and tracked. In addition, a few keyframes were required to make the tracker more robust. L. Vacchetti et al. pointed that it was very convenient

to estimate the camera position from a single image in order to initialize the tracker and to recover the failure automatically. Active Appearance Models based approaches [9, 10, 35] provide a good solution to recover the 2D affine pose parameters along with the feature points from single image. Recently, researchers [29, 1, 41] have attempted to build the AAM with three dimensions.

P. Mittrapiyanumic [29] proposed two AAMs algorithms for rigid object tracking and pose estimation. The first method is to utilize two instances of AAM to track landmark points in a stereo pair of images and perform 3D reconstruction of the landmarks followed by 3D pose estimation. The second method, i.e., AAM matching algorithm, is an extension of the original AAM that incorporates the full six degrees of freedom pose parameters as part of the parameters for the minimization. The results showed that the accuracy in pose estimation of appearance based methods is better than the methods using the geometric approach. J. Ahlberg [1] proposed an approach using the 3D AAM for face and facial feature tracking, in which the depth information of 3D shape was acquired by fitting a generic model. In addition, the pose parameters were estimated from a motion tracker, then the shape model parameters were recovered by AAM fitting.

Jing Xiao et al. [41] proposed a non-rigid structure-from-motion algorithm that could be used to convert a 2D AAM into a 3D face model. They then described how a non-rigid structure-from-motion algorithm was able to be employed to compute the corresponding 3D shape models from a 2D AAM. Their method did not require 3D range data in [4] and also shared fast fitting speeds. They then showed how the 3D modes could be used to constrain the AAM so that it could only generate model instances, but also could be generated with the 3D modes. Their fast fitting algorithm mainly benefited from the projection-out method and Inverse Compositional update strategy, thus the Jacobi matrix was constant. However, the approximation that the shape Jacobi matrix was made orthogonal to the texture Jacobi matrix, was only valid for few texture modes. Only shape parameters were recovered iteratively, and the texture parameters were recovered linearly in one step. In addition, the recovered pose parameters were not accurate enough, mainly because the pose parameters were compensated by the shape variations. A weak perspective camera model was employed in order to decrease the computational cost, and the full perspective camera model was necessary for the common AR applications. These may limit their applications particularly for AR applications.

2.2 Overview

Here is the overview of the proposed approach. First, we construct 3D shape models from AAMs offline, which do not involve processing of the 3D scan data. Based on the computed 3D shape models, we propose an efficient online algorithm to estimate both 3D pose and non-rigid shape parameters via local bundle adjustment

for building up point correspondences. Our approach, without manual intervention, can recover the 3D non-rigid shape effectively from either real-time video sequences or single image. The recovered 3D pose parameters can be used for AR registrations. Furthermore, the facial feature can be tracked simultaneously, which is critical for many face related applications. We evaluate our algorithms on several video sequences. Promising experimental results demonstrate our proposed scheme is effective and significant for real-time AR applications.

In this chapter, we present a novel scheme of real-time non-rigid shape recovery via active appearance models for augmented reality applications. The rest of this chapter is organized as follows. Section 2 reviews the AAM algorithm and describes an extended AAM matching algorithm which predicts shape directly from texture for improving the accuracy of AAM searching. Section 3 presents our proposed scheme. We first provide an overview of our scheme in the context of augmented reality applications in Section 3.1. Then Section 3.2 describes how to construct the 3D shape models based on the 2D AAM tracking results. Section 3.3 presents a novel and efficient algorithm for online estimation of 3D pose and non-rigid shape parameters simultaneously via local bundle adjustment. Section 3.4 gives our experimental results and the details of our experimental implementation. Section 4 discusses the critical requirements of real-time AR applications, several major differences of our proposed scheme compared with previous work, and the advantages of our scheme particularly for AR applications as well as the disadvantages and our future work. Section 5 sets summarizes this chapter.

2.3 An Extended AAM Matching Algorithm

The Active Appearance Models (AAMs) [9, 10, 35, 1] have been proven as a successful method for matching statistical models of appearance to new images. AAMs are taking the analysis-through-synthesis approach to the extreme. This approach has been successfully applied in numerous different applications. AAMs establish a compact parameterizations of object variability, as learned from a training set by estimating a set of latent variables. The modelled object properties are usually shape and pixel intensities. There are several modifications for the basic AAM algorithm [10]. One approach was the Direct Appearance Model (DAM) for improving the convergence speed and searching accuracy by predicting the shape directly from the texture [17]. In the following, we first review the background of the original AAM approach and the improved DAM method. Based on the DAM method, we describe an extended algorithm for AAM matching in our scheme.

The algorithm of AAM Matching

1. Generate texture vector $\mathbf{g}_m$ from model
2. Sample image below the model shape $\mathbf{g}_i$
3. Evaluate error vector $\mathbf{r} = \mathbf{g}_i - \mathbf{g}_m$ and error $\mathbf{E} = |\mathbf{r}|$
4. Compute displacements in pose $\delta\mathbf{t} = \mathbf{R_t r}$
5. Compute displacements in texture $\delta\mathbf{b_t} = \mathbf{R_g r}$
6. Update pose and texuture parameters with initial $k = 1$
7. Transform the shape by the estimated parameters
8. Repeat step 1-3 to form a new error $\mathbf{E}^{'}$
9. If $\mathbf{E}^{'} < \mathbf{E}$ accept the new estimate,
   otherwise goto step 6 to try other k=0.5, 0.25, ...

Figure 2.1. An extended AAM matching algorithm

2.3.1 Active Appearance Model

In the AAMs [10], the appearance model is defined by its shape and texture. The shape model is a set of 2D landmark points represented as a vector $\mathbf{s}$:

$$\mathbf{s} = (x_1, ..., x_n, y_1, ..., y_n)^{\mathbf{T}} . \tag{2.1}$$

where (x_i, y_i) are the coordinates of the i^{th} landmark point. The texture model is defined by the set of intensity values of the pixels lying inside the reference shape. These intensity values form a vector $\mathbf{t}$, which represents the texture model.

The 2D shape and texture are controlled by a statistical model. They can be represented as a base plus a linear combination of variations:

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P_s b_s} \tag{2.2}$$

$$\mathbf{t} = \bar{\mathbf{t}} + \mathbf{P_t b_t} \tag{2.3}$$

where $\mathbf{b_s}$ are 2D shape parameters, and $\mathbf{b_t}$ are the texture parameters.

The combined appearance model contains parameters, $\mathbf{c}$, controlling the 2D shape and texture at the same time. Thus, the new object instance can be generated by

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P_s W_s^{-1} P_{cs} c} \tag{2.4}$$

$$\mathbf{t} = \bar{\mathbf{t}} + \mathbf{P_t P_{ct} c} \tag{2.5}$$

where $\mathbf{W_s}$ is a diagonal weight matrix between pixel distances and pixel intensities. A simple choice of the weight matrix is to set $\mathbf{W_s} = \mathbf{wI}$, where $\mathbf{w}^2$ is the ratio of the total texture intensity variation to the total 2D shape variation.

2.3.2 Direct Appearance Model

Hou et al. [17] suggested that in some cases it is possible to predict the 2D shape directly from the texture, when two are sufficiently correlated. From (4) and (5),

$$\mathbf{b_s} = \mathbf{P_{cs} P_{ct}^{+} b_t} \tag{2.6}$$

where $\mathbf{P_{ct}^{+}}$ is the pseudo inverse of $\mathbf{P_{ct}}$. Thus, the 2D shape parameters from the texture can be predicted accurately if the rank of $\mathbf{P_{ct}}$ is larger than the number of 2D shape parameters. Hou et al. [17] demonstrated that this is the case for a face model.

Let $\mathbf{Q_s} = \mathbf{P_s W_s^{-1} P_{cs} P_{ct}^{+}}$, the direct appearance model instance can be generated by

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{Q_s b_t} \tag{2.7}$$

$$\mathbf{t} = \bar{\mathbf{t}} + \mathbf{P_t b_t} \tag{2.8}$$

2.3.3 An Extended AAM Matching Algorithm

The AAM matching algorithm tries to minimize the residual between the model and image $\mathbf{r} = \mathbf{g}_i - \mathbf{g}_m$, where $\mathbf{g}_i$ is the sampled image below model shape, and $\mathbf{g}_m$ is the model texture. During the DAM training phase, one learns the relationships

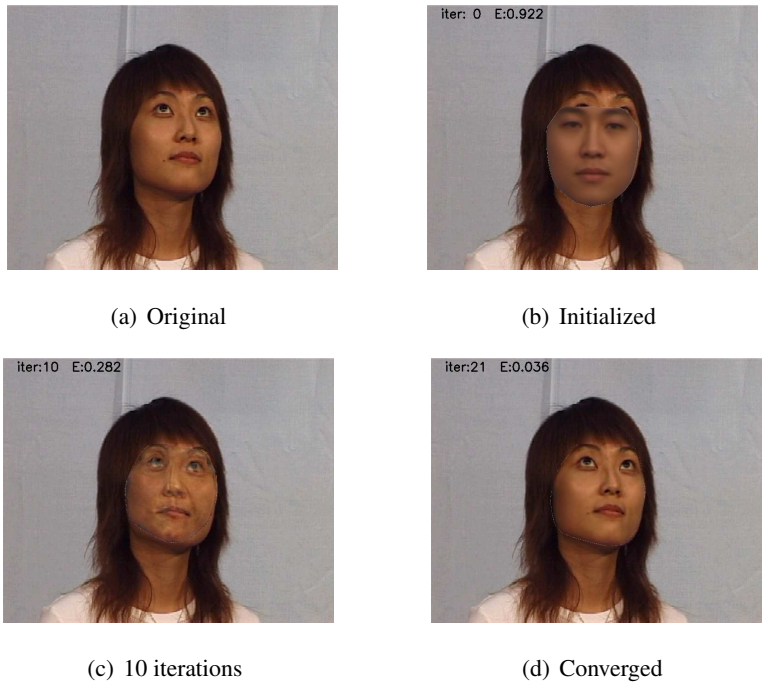
$$\delta \mathbf{t} = \mathbf{R_t r} \ ,$$

$$\delta \mathbf{b_t} = \mathbf{R_g r} \ .$$

Instead of using a traditional approach for AAM matching in [9], we implement a modified AAM fitting algorithm for quicker convergency and better matching accuracy similar to the approach in [35]. The proposed iterative AAM matching algorithm which predicts shape directly from texture is given in Fig. 2.1.

In our experiments, the AAMs are built up with 140 still face images belonging to 20 individuals, seven images for each. Each image is manually labelled with 100 points. As shown in Fig. 2.2, the matching experiment is performed on an AAM with 14 shape parameters, 68 texture parameters, and 36335 color pixels. Fig 2.2

respectively show (a) the original single image, (b) the initialization of our AAM fitting, (c) the result after 10 iterations and the final converged result after 21 iterations. In each case the rendered model images and estimation errors are displayed in the figures.



(a) Original (b) Initialized

(c) 10 iterations (d) Converged

Figure 2.2. An example of our AAM fitting to a single image. The estimated errors are displayed in each case.

2.4 Real-Time Non-Rigid Surface Recovery for AR

2.4.1 Overview of Our Solution

Our scheme tries to attack the critical problems of pose and non-rigid shape recovery. Traditional techniques may be neither flexible and powerful enough for model representations nor efficient enough for real-time purposes. For tackling the challenges, we solve the problem by a two-stage scheme via AAM techniques:

- We acquire the 2D shape of objects using the AAM fitting algorithm described in Section 2.3 firstly, then construct the 3D shape basis offline based on the AAM fitting results.
- We estimate the 3D pose and 3D shape parameters online simultaneously via local bundle adjustment by building up the point correspondences between 2D and 3D.

The above proposed solution differs from the regular approach in [38] which estimated the pose of an object through point matching. To exploit the representational power of AAMs, instead of matching points between two frames, we propose a novel approach to setup the point correspondences between the 2D and 3D shape via AAM fitting to a single image. This procedure needs no manual initialization. The details of our approach are described as follows.

2.4.2 Offline Construction of 3D Shape Basis

Bregler et al. [6] proposed a solution for recovering 3D non-rigid shape models from image sequences. Their technique is based on a non-rigid model, where the 3D shape in each frame is a linear combination of a set of basis shapes. By analyzing the low rank of the image measurements, they proposed a factorization-based method that enforces the orthonormality constraints on camera rotations for reconstructing the non-rigid shape and motion. Torresani et al. [36] extended the method in [6] to initialize the optimization process. By using the extended AAM matching algorithm in Section 2.3, we first acquire the 2D shapes of objects. With the trained 2D shapes, we are able to construct the 3D shape basis due to the powerful representational capability of AAMs [41].

The 3D shape can be described as a set of key-frame basis $S_1, S_2, \cdots, S_m$. Each key-frame S_i is a $3 \times n$ matrix. The 3D shape of a specific configuration is a linear combination of the following basis set:

$$\mathbf{S} = \mathbf{S_0} + \sum_{i=1}^{m} p_i \mathbf{S_i} \qquad \mathbf{S}, \mathbf{S_i} \in R^{3\times n}, p_i \in R \tag{2.9}$$

where the coefficients p_i are the 3D shape parameters, and $\mathbf{S_i}$ are the 3D coordinates: $\mathbf{S} = \{\mathbf{M_1}, \mathbf{M_2}, \cdots, \mathbf{M_n}\}, \mathbf{M_i} \in R^{3\times 1}$. Under a weak perspective projection, the n points of $\mathbf{S}$ are projected into 2D image points (u_i, v_i):

$$\begin{bmatrix} u_1 & u_2 & \cdots & u_n \\ v_1 & v_2 & \cdots & v_n \end{bmatrix} = \mathbf{R} \cdot (\sum_{i=0}^{m} p_i \mathbf{S_i}) + \mathbf{T} \tag{2.10}$$

$\mathbf{R}$ contains the first 2 rows of the full 3D camera rotation matrix, and $\mathbf{T}$ is the camera translation. The scale of the projection is coded in $p_1, p_2, \cdots, p_m$. The camera translation $\mathbf{T}$ is eliminated by subtracting the mean of all 2D points, and henceforth one can assume that $\mathbf{S}$ is centered at the origin.

If the AAMs are tracked through a sequence of N images, 2D points of the AAM shape in each frame can be obtained. Let us add a temporal index to each 2D point, and denote the tracked points in frame t as (u_i^t, v_i^t). All points of AAM shape in all N images are stacked into one large measure $2N \times n$ matrix W. The number of 3D

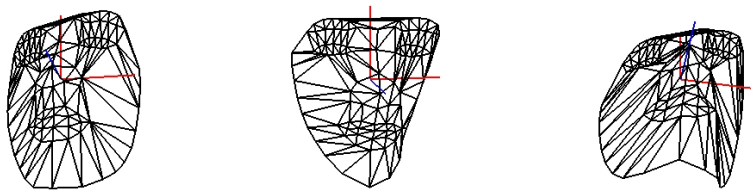
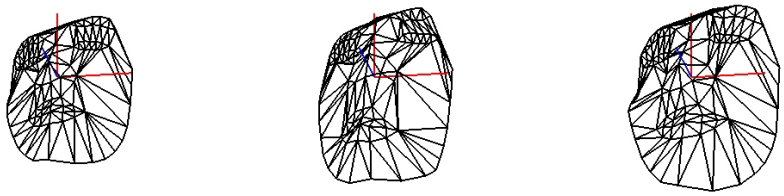
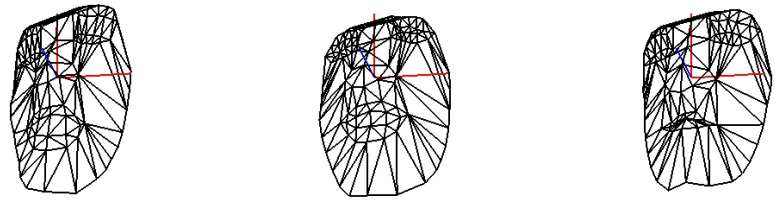


Figure 2.3. An example of 3D mean shape of three views $\mathbf{S_0}$



(a) $\mathbf{S_1}$ (b) $\mathbf{S_2}$ (c) $\mathbf{S_3}$



(d) $\mathbf{S_4}$ (e) $\mathbf{S_5}$ (f) $\mathbf{S_6}$

Figure 2.4. An example of the first six 3D shape modes (a-f) from an AAM.

shape verities equals to the number of 2D AAM vertices n, it can be rewritten as follows:

$$W = \begin{bmatrix} u_1^1 & u_2^1 & \cdots & u_n^1 \\ v_1^1 & v_2^1 & \cdots & v_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ u_1^N & u_2^N & \cdots & u_n^N \\ v_1^N & v_2^N & \cdots & v_n^N \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{R}_1 & p_1^1\mathbf{R}_1 & \cdots & p_m^1\mathbf{R}_1 \\ \mathbf{R}_2 & p_1^2\mathbf{R}_2 & \cdots & p_m^2\mathbf{R}_2 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{R}_N & p_1^N\mathbf{R}_N & \cdots & p_m^N\mathbf{R}_N \end{bmatrix}}_{\mathbf{M}} \cdot \underbrace{\begin{bmatrix} \mathbf{S_0} \\ \mathbf{S_1} \\ \vdots \\ \mathbf{S_m} \end{bmatrix}}_{\mathbf{B}} \tag{2.11}$$

where $\mathbf{M}$ is a $2N \times 3(m+1)$ scaled projection matrix and $\mathbf{B}$ is a $3(m+1) \times n$ shape matrix. In the noise-free case, $\mathbf{W}$ has a rank $r \leq 3(m+1)$, which can be factorized into the product of a $2N \times 3(m+1)$ matrix $\tilde{\mathbf{M}}$ and a $3(m+1) \times n$ matrix $\tilde{\mathbf{B}}$. This decomposition is not unique, which can be determined by a linear transformation. Any non-singular $3(m+1) \times 3(m+1)$ matrix $\mathbf{G}$ and its inverse could be inserted between $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{B}}$. In addition,

their product still remains equal to $\mathbf{W}$. Namely, we have the following equations

$$\mathbf{M} = \tilde{\mathbf{M}} \cdot \mathbf{G} \tag{2.12}$$

$$\mathbf{B} = \mathbf{G}^{-1} \cdot \tilde{\mathbf{B}} \tag{2.13}$$

where the corrective matrix $\mathbf{G}$ can be found by solving a least square optimization problem [6]. Thus, given 2D tracking data $\mathbf{W}$, a non-rigid 3D shape matrix with r degrees of freedom can be estimated, along with the corresponding camera rotations and configuration weights for each time frame.

In our experiments, we implement the AAM matching algorithm given in Section 2.3 and run it to fit the short video sequences of of 20 individuals (2678 frames in total). The training results are employed to construct the 3D shape basis in our experiments. Fig. 2.3 shows an example of the computed 3D mean shape modes of three views from AAM. Fig. 2.4 shows the first six 3D shape modes from an AAM.

2.4.3 Real-Time Non-Rigid Surface and Pose Recovery for AR

To make it flexible and general for wide applications, we employ the perspective camera model, in which a 3D point $\mathbf{Q}$ is re-projected based on the 2D point $\mathbf{q}$:

$$\mathbf{q} = \mathbf{A}[\mathbf{R}|\mathbf{T}] \cdot \mathbf{Q}$$

where the camera rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{T}$ estimated from the current frame are expressed in the object coordinate system, and $\mathbf{A}$ is the intrinsic camera matrix. The intrinsic parameters of the camera can be calculated offline. This does not require to be done precisely, and typically an approximate configuration is sufficient. Hence, we can assume the intrinsic parameters are fixed. Moreover, in order to allow some deformation, the rigid shape model is replaced by the 3D linear shape model. We now describe how to in real-time estimate the 3D pose parameters and non-rigid shape parameters simultaneously.

Given the constructed 3D shape basis via AAM training algorithm, we can build up the 2D-3D correspondences. Based on the established correspondences, an efficient way for estimating the parameters of camera position and the 3D shape coefficients can be turned into minimizing the re-projection error:

$$\min_{\mathbf{R},\mathbf{T},p} \rho\left(\mathbf{s}, \phi\left(\mathbf{A}[\mathbf{R}|\mathbf{T}], \mathbf{S}\right)\right) \tag{2.14}$$

Let $\mathbf{S} = \mathbf{S_0} + \sum_{i=1}^{m} p_i \mathbf{S_i}$, the optimization problem can be written as

$$\min_{\mathbf{R},\mathbf{T},p} \rho\left(\mathbf{s}, \phi\left(\mathbf{A}[\mathbf{R}|\mathbf{T}], \mathbf{S_0} + \sum_{i=1}^{m} p_i \mathbf{S_i}\right)\right) \tag{2.15}$$

with respect to the orientation and translation parameters $\mathbf{R}$ and $\mathbf{T}$, where

- ρ is the robust M-estimator [27] in consideration of outliers which can be given as follows:

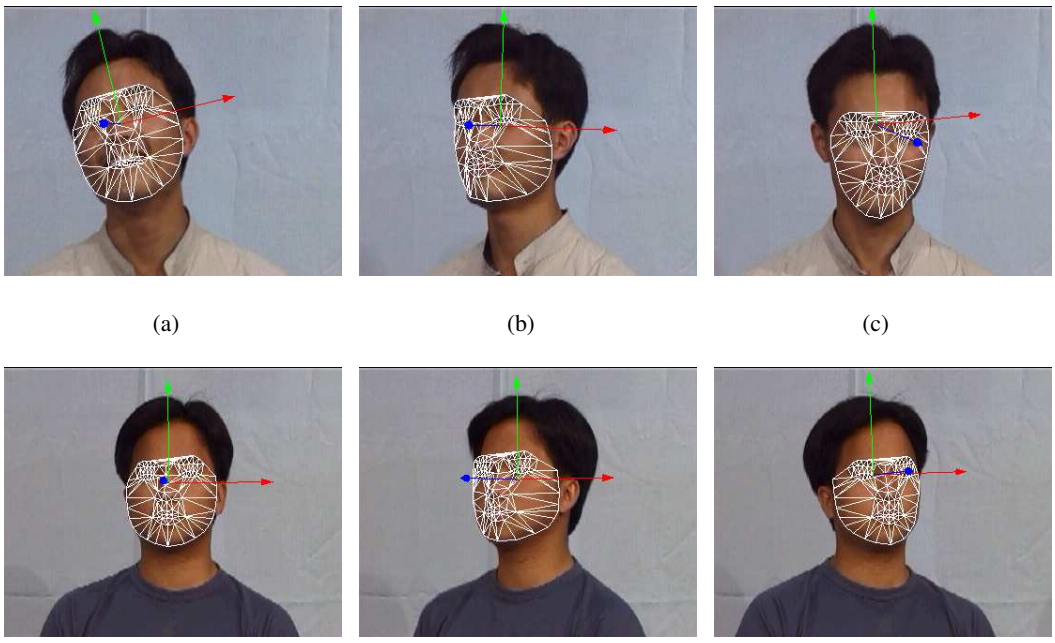
$$\rho(u) = \begin{cases} \frac{\alpha^2}{6}[1-(1-(\frac{u}{\alpha})^2)^3], & |u| \leq \alpha \\ \frac{\alpha^2}{6}, & |u| > \alpha \end{cases} \tag{2.16}$$

- $\phi\left(\mathbf{A}[\mathbf{R}|\mathbf{T}], \mathbf{S_0} + \sum_{i=1}^{m} p_i \mathbf{S_i}\right)$ denotes the projection of 3D shape given the parameters $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{T}$.

The above optimization procedure can converge quickly within a couple of iterations when it begins with a good initial estimation.

2.5 Experimental Results

2.5.1 3D Face Tracking and Augmented Reality



(a) (b) (c)

(d) (e) (f)

Figure 2.5. Tracking faces using proposed method in the augmented video sequences, the axis in the displayed frames indicates the current 3D pose of tracked subject.

The results of estimated 3D shapes of two individuals are depicted in Fig. 2.5, which are extracted from two video clips with total 300 frames. We can see that the 3D shapes are successfully fitted to the face image. The face deformation can be well described by 6 3D shape parameters, for example, fitting to different individuals with the same AAM model in Fig. 2.5(a-f). The algorithm can handle large pose variations and displacements, as shown in

Fig. 2.5(a,b,e,f). Fig. 2.5(a,c) revealed that the proposed approach can handle tilt pose, and Fig. 2.5(d-f) displayed the results which deal with out-of-plane rotation. In each result image, the axis indicates the current orientation and translation. Since the intrinsic and extrinsic camera matrices are computed, the virtual rigid and deformable objects can be inserted into the scene. Fig. 2.6 shows that a rigid virtual glasses and a deformable beard are added into the video sequences. From the results, we can observe that the beard can be deformed along with the expression changes. The added virtual objects are tightly overlaid on the subject. We use the results of previous frames as the initial values for the optimization, thus, only 3-4 iterations per frame required for AAM convergence. Since no relation with image information, the 3D pose and 3D shape parameters are computed efficiently.

(a) (b) (c)

Figure 2.6. Adding glasses and beard to the subject in the augmented video sequence, the beard is deformed along with the expression changes.

Fig. 2.7 plots the re-projection error in the online non-rigid shape recovery step when varying number of 3D shape basis $\mathbf{m}$. The experiment is performed on a video clip with 65 frames. As shown in Fig. 2.7, large error occurred only rigid shape is used for pose estimation, and the error reaches $700/100 = 7$ per point. The re-projection error decreases significantly when introducing the 3D linear shape model, additionally, it becomes smaller when $\mathbf{m}$ grows up. When six 3D shape basis are used, the average re-projection is below $100/100 = 1$ each point. However, large number of nonlinear parameters would affect the convergence speed of the object function, there is a trade-off between the accuracy and efficiency. Furthermore, large number of 3D shape basis may decreases the number of optimization iterations.

2.5.2 Re-texturing Faces

We also apply the proposed method for re-texturing a human face. The scheme of the whole system is shown in Fig. 2.8. In order to demonstrate our proposed nonrigid shape and pose recovery approach is effective and promising for generating novel view and 3D facial animation purposes, we first map the recovered 3D nonrigid

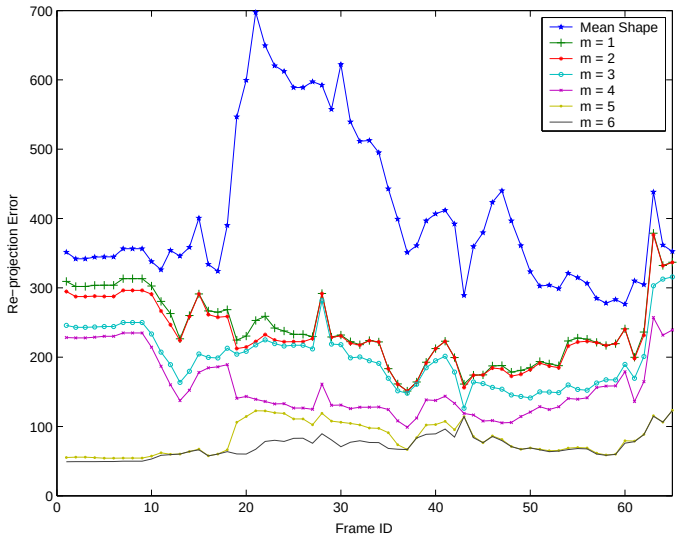


Figure 2.7. The re-projection error with various number of 3D shape basis

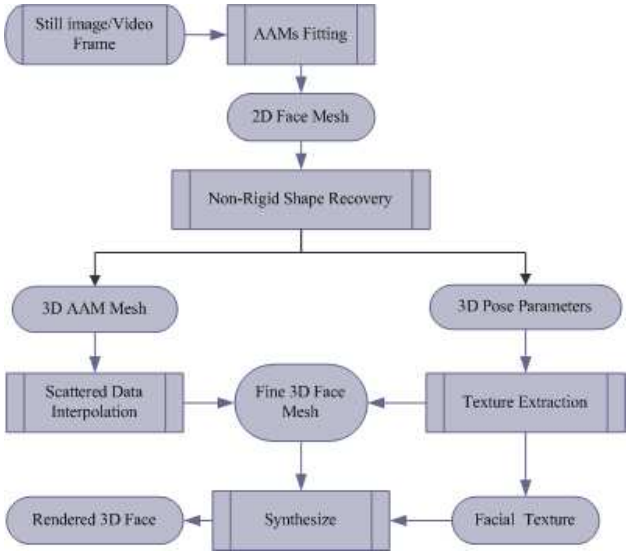


Figure 2.8. Overview the method for re-texturing a face

shape into high resolution mesh via interpolation [30], then render the novel views by mapping different texture and with different poses.

Scattered Data Interpolation

As non-rigid 3D face shape and pose are computed from the 2D shape fitted by the AAMs, the 3D facial feature points are extracted. We can use these points to deform a generic 3D mesh with a large number of vertices. A smooth interpolation function gives the 3D displacements between facial point positions and the newly adapted positions for every vertex in the target mesh. Constructing such an interpolation function is a standard problem in scattered data interpolation [30]. Given a set of known displacements $u_i = p_i - p_i^0$ away from the original position p_i^0 at every constrained vertex i, we can construct a function that gives the displacement u_j for every unconstrained vertex j. A smooth vector-valued function $f(p)$ is fitted to the known data $u_i = f(p)$, from which the unknown data u_j can be computed via $u_j = f(p)$. In [32], a function is engaged:

$$
\begin{aligned}
f(p) &= \sum_i \mathbf{c_i} \phi(||p - p_i||) \\
s.t. \quad & \sum_i \mathbf{c_i} = 0 \\
& \sum_i \mathbf{c_i} p_i^{\mathbf{T}} = 0
\end{aligned}
$$

where $\phi(r) = e^{(-r/256)}$ are radially symmetric basis functions.

Texture Extraction

Extracting the texture maps is necessary for rendering photo-realistic images of a reconstructed face model for various viewpoints [15]. After the pose and 3D face mesh are computed, we use an approach similar to the method [32] to generate a texture map

Results

Fig. 2.9 shows rendered enlarged novel view rotated from the current pose by 20° on Y-axis.

Fig. 2.10 shows the experimental result by replacing face texture of a person with anther person. In the Fig. 2.10, the top left one is the modelled person and the bottom left is the constructed 3D mesh in which 3D pose information is available; the top right one is the front face of the replaced person and the bottom right shows the generated results by replacing the texture using the built 3D model and 3D pose parameters. The generated view fits well on the 3D model. But one may find the skin is not smooth since we do not consider the lighting condition; this can

Figure 2.9. The synthesis result using different facial texture

be easily improved by adding smooth operations and lighting adjustment. But the experimental results can answer our question that our constructed 3D model are effective and promising for 3D facial animations.

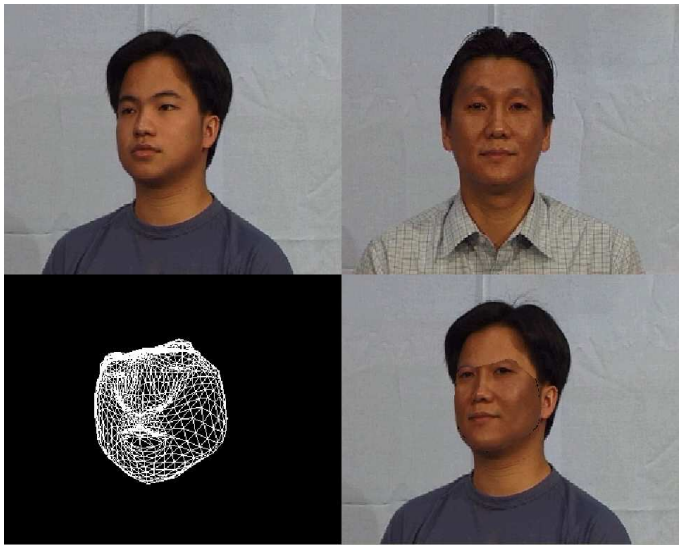


Figure 2.10. The synthesis result using different facial texture

2.5.3 Computational Complexity

We evaluate the computational cost of the proposed method on a Pentium III 1GHz CPU. It runs at 200ms per image of size 352×288. AAM fitting takes 40ms and 3D recovery step takes 74ms. The AAM with 10 shape parameters, 52 texture parameters. The non-rigid shape recovery step with 6 camera parameters and 6 3D shape parameters.

2.6 Discussions

In this section, we discuss several major differences and advantages of our proposed scheme compared with previous work from several aspects in which we show that our proposed scheme is particularly flexible and powerful for augmented reality applications. Finally, we also mention the disadvantages and some improvements in our future work.

Rigid vs. Non-Rigid. The prior model employed by L. Vacchetti et al. [38] is only for rigid objects or deformable objects with small variations. P. Mittrapiyanumic et al. [29] do not take full advantage of AAM's deformation power, the AAM is just used to estimated the 3D pose of rigid objects. The proposed method can deal with 3D deformation through introducing 3D linear shape models. In addition, large variation can be obtained by increasing the number of 3D shape basis. The facial feature can be located accurately by the power of AAM fitting, thus, the added virtual beard can be deformed with the facial expressions in Fig. 2.6. The novel view can be generated from the current view, even the facial texture of different individuals can be exchanged, as shown in Fig. 2.10. Additionally, the proposed approach provides a solution for building the 2D-3D correspondence from single image. Thus, the tracker can be initialized without manual intervention. In addition, the failure can be recovered automatically.

Offline vs. Online. Many methods [6, 36] have been presented for offline non-rigid shape recovery from image sequences through factorizing analysis on the 2D tracked points. Different from these approaches, our proposed method is able to work online by exploiting the 3D shape models that can be constructed offline effectively by using AAM tracking. This enables us to online acquire 3D non-rigid shape and pose which can be applicable for many AR applications.

Advantages for AR applications. In [41], the model and the fitting algorithm are person specific. The generic AAM is slower than the person specific AAM, but provides good accuracy in the case of large texture variations[14]. In addition, the inverse compositional update strategy is good for smooth shape, and not for the non smooth ones. The proposed extended AAM is a generic model with additive update method rather than person specific model with inverse compositional approach. Thus, it can handle large texture variations, fitting to different individuals. The weak-perspective model used in "Combined 2D+3D AAM" is not suitable for augmented reality applications, moreover, the optimization procedure of the algorithm is complicated. We optimize AAM and 3D pose parameters respectively. Virtual objects can be added to the scene by the estimated camera, orientations and translations information. In addition, the proposed approach is more flexible. The AAM fitting step can be replaced with other algorithms, such as Active Shape Models based approaches [27].

Disadvantages and Future Work. The proposed approach does not take full advantage of 3D information for speeding up AAM convergence. The accuracy of AAM fitting is critical to the 3D pose output. Large rotation may be compensated by the 3D linear mode, therefore, the estimated pose is not so accurate. In the future, problem mentioned above will be solved by training the 3D AAM with the aligned 3D shapes instead of 2D shapes.

2.7 Summary

In this chapter we presented a novel scheme for non-rigid shape recovery in real-time augmented reality applications. Our scheme first builds the 3D shape models offline using an effective AAM algorithm. Given the constructed 3D shape models, an efficient online algorithm is suggested to estimate both the 3D pose and non-rigid shape parameters simultaneously. One of our main contributions is the introduction of 3D linear shape model to estimate the 3D pose parameters and non-rigid shape simultaneously via local bundle adjustment. Moreover, we suggested an updating scheme to predict the shape directly from texture that can improve the accuracy of AAM searching. The promising experimental results validate our proposed scheme is effective for real-time AR applications.

Chapter 3

Progressive Finite Newton Approach

In this chapter, we propose a novel progressive finite Newton optimization scheme for the nonrigid surface detection problem, which is reduced to only solving a set of linear equations. We have conducted extensive experiments for performance evaluation on various environments, whose promising results show that the proposed algorithm is more efficient and effective than the existing iterative methods.

3.1 Motivation

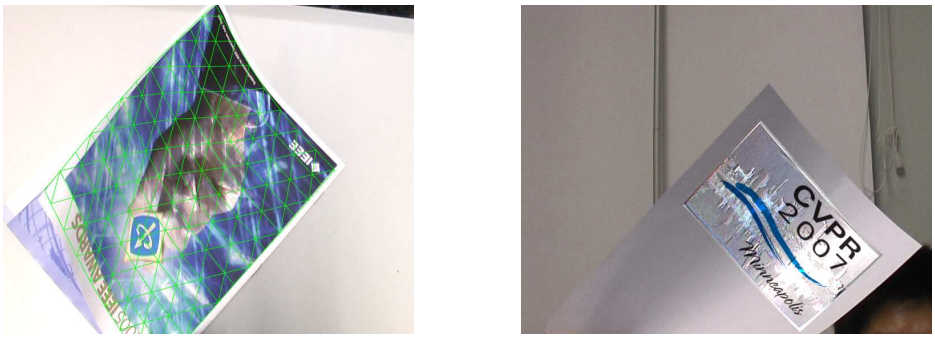
The detection and tracking of the nonrigid objects in images and videos is an interesting and beneficial research issue for computer vision and image analysis [2, 33, 37]. The goal of nonrigid surface detection is to extract the deformable shape's structure from an input image. The difference between nonrigid surface recovery and detection is that the latter does not require any initialization or a priori pose information. An effective nonrigid surface detection technique can be applied in a variety of applications for digital entertainment, medical imaging [2] and augmented reality, such as the re-texturing of images and videos [39, 40].

Nonrigid surface detection can usually be treated as the problem of recovering the explicit surface with a few deformation parameters and finding out the correct correspondences from noisy data simultaneously. Many applications have been investigated for deformable object tracking and registration, such as face tracking and modelling [9, 41, 42, 4, 11], and also more generic and more deformable objects [2]. The major problem of these methods is that they tend to be computationally expensive and mainly aim at object recognition and image segmentation tasks rather than nonrigid surface recovery. However, a real-time and automated solution [33] has recently been proposed, which takes advantage of an iterative robust optimization scheme.

Unlike the rigid object pose estimation, it is difficult to directly employ a robust estimator, such as RANSAC [12] or Hough transform [16], to remove the spurious matches for nonrigid surface detection. Because the nonrigid

surface is usually highly dynamic and represented by many deformation parameters, the problem is far more complex than the rigid object detection. Moreover, it requires a sufficient number of correct correspondences in order to obtain high registration accuracy. An alternative strategy is to iteratively solve for both the correspondence and the transformation [3, 33]. However, these methods are either sensitive to initial conditions and parameter choices, or involve too many iterations and a complex optimization procedure. Consequently, they are neither efficient nor effective for real-time applications.

(a) Starbucks pad (b) T-shirt



(c) Cover (d) Paper

Figure 3.1. Detecting nonrigid surfaces in real-time video (a-d). (a) The contour is overlaid on the Starbucks pad. (b) T-shirt with shadow. (c) The cover of a magazine. (d) A piece of paper with specular reflection.

3.2 Overview

In this chapter, we propose a novel progressive finite Newton optimization scheme for nonrigid surface detection, which has the advantage of solving only a fixed number of linear equations. Moreover, a progressive

sample scheme far more efficient than RANSAC is proposed to initialize the optimization process. The previous method [33] is currently generally accepted as the most effective methods of solving this kind of problem. It employs an implicit iterative scheme for the first order partial differential equation; however, this requires a large number of iterations to solve the problem and remove the outliers simultaneously. We tackle this critical problem from two angles. First, the nonrigid surface detection is formulated as an unconstrained quadratic optimization problem, which has a closed-form solution for a given set of observations. Thus, it can be efficiently solved through LU factorization. Then, a progressive sample [8] scheme is employed to initialize the optimization scheme, which can decrease the number of trials significantly. Therefore, the present approach requires much fewer iterations than the semi-implicit iterative optimization scheme [34]. Thus, the proposed method is very efficient for real-time nonrigid surface recovery tasks. To evaluate the performance of our proposed algorithm, we conduct extensive experiments on such diverse objects as a Starbucks pad, a T-shirt, and the cover of a magazine, as shown in Fig. 3.1.

The rest of this chapter is organized as follows. Section 2 reviews the previous approaches employed for the nonrigid surface detection and recovery. In Section 3, we present the proposed progressive finite Newton solution. Section 3.1 describes the nonrigid surface model and mapping function for a feature matching-based method. Section 3.2 presents the object function which minimizes the correspondence errors and surface energy. A robust estimator is introduced to deal with the large outliers. In Section 3.3, the nonrigid surface detection is formulated as an unconstrained quadratic optimization problem, which is efficiently solved using the factorization method. Section 3.4 presents the progressive finite Newton optimization scheme to remove the spurious correspondences, and the progressive sampling method to initialize the optimization. Section 4 provides the details of our experimental implementation and describes our experimental results. We discuss limitations and future work in Section 5. Section 6 summarizes this chapter.

3.3 Related Work

Although nonrigid surface detection in general is not new to researchers in the computer vision domain, only a few approaches are automatic and can achieve real-time results. Some appearance-based approaches directly minimize the residual image between the input image and the synthesized model image [9]. Moreover, optical flow information [11, 2] can be incorporated into the optimization scheme to obtain better results. However, the major limitation of these methods is that they tend to become stuck at a local minimum and hence require good initialization. In addition, it is usually difficult to handle the partial occlusion for an appearance-based method. Well-designed markers widely used in motion capture are also applied to recover the structure of a

nonrigid surface, such as cloth and paper [40, 39]. As these methods rely on the physical markers, they require the placing of pre-defined patterns on the target surface. Nevertheless, they are capable of high accuracy. On the other hand, feature-based methods [3, 33] try to find out the transformation from the correspondences built by feature matching methods. Thus, these methods can benefit from the recent advances in the feature detection and matching. In [33, 34], J. Pilet et al. proposed an iterative approach to attack the fast nonrigid surface recovery problem. Physical constraints based on the Finite Element Model [37] are employed for regularization. A semi-implicit iterative scheme is proposed to solve the optimization problem.

Recently, several sophisticated feature descriptors [25, 28] have been proposed to handle the wide-baseline matching problem, including images with large deformation [24]. In addition, machine learning methods, such as random classification trees [23], are also employed to find the point correspondences. These methods can take advantage of shifting part of the computational load from the matching phase to the training phase.

It is more complex to handle a large amount of deformation parameters for detecting the nonrigid surface rather than only a few pose parameters used in rigid object detection. Therefore, there are several challenges when applying conventional robust estimators, such as RANSAC and M-estimator, for the nonrigid surface detection task. One is the lack of a concise function which can estimate the deformed mesh from the correspondences directly; instead, one may need to use a large number of free variables, which can lead to a high computational cost for each prediction step. Obviously, the semi-implicit iterative approach [33] is not efficient enough to deal with this problem. Another challenge is that the RANSAC-based approach requires a large number of trials. This makes the problem even more complex. Moreover, to the best of our knowledge, there is still a lack of criteria for selecting the number of samples for each trial in nonrigid surface detection. In rigid object pose estimation, the sample number is usually set according to the number of free parameters. However, the number of deformation parameters for a nonrigid surface may be larger than the total number of observations. We tackle the initialization problem through a modified RANSAC method. The key is to draw from progressively larger sets of top-ranked correspondences [8], rather than to treat all correspondences as equal and draw random samples uniformly from the full set in RANSAC. Thus, our progressive sample scheme affords large computational savings, and the conventional robust estimator can be engaged for initializing the nonrigid surface detection.

In contrast to the previous work, our proposed approach is based on a progressive finite Newton scheme, in which the optimization problem can be solved very efficiently by a factorization method. In addition to offering computationally highly competitive performance, our proposed modified RANSAC initialization method can further reduce the number of Newton optimization steps.

3.4 Nonrigid Surface Detection

In this section, we describe the present progressive finite Newton optimization scheme for detecting and recovering the nonrigid surface. For tackling the challenges, a mapping function is used to associate the feature correspondences with a mesh model. Therefore, the nonrigid surface detection turns out to be a problem which minimizes the correspondence error and the surface energy. Moreover, we formulate the nonrigid surface detection into an unconstrained quadratic optimization problem. A progressive scheme is proposed to deal with outliers and find out as many correct correspondences as possible. Finally, a modified RANSAC scheme is introduced to select the initial active set for the optimization scheme.

3.4.1 2D Nonrigid Surface Model

The nonrigid surface is usually explicitly represented by triangulated meshes. As shown in Fig. 3.1(c), we employ a triangulated 2D mesh with N hexagonally connected vertices, which are formed into a shape vector $\mathbf{s}$ as below:

$$\begin{aligned} \mathbf{s} &= \begin{bmatrix} \mathbf{x} & \mathbf{y} \end{bmatrix}^\top \\ &= \begin{bmatrix} x_1 & x_2 & \dots & x_N & y_1 & y_2 & \dots & y_N \end{bmatrix}^\top \end{aligned}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the vectors of the coordinates of mesh vertices. We assume that a point $\mathbf{m}$ lies in a triangle whose three vertices' coordinates are (x_i, y_i),(x_j, y_j) and (x_k, y_k) respectively, and $\{i, j, k\} \in [1, N]$ is the index of each vertex. The piecewise affine transformation is used to map the image points inside the corresponding triangle into the vertices in the mesh. Thus, the mapping function $T_S(\mathbf{m})$ is defined as below

$$T_{\mathbf{s}}(\mathbf{m}) = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \end{bmatrix}^\top \tag{3.1}$$

where (ξ_1, ξ_2, ξ_3) are the barycentric coordinates for the point $\mathbf{m}$.

3.4.2 Nonrigid Surface Recovery

In general, the nonrigid surface detection problem approximates a 2D mesh with $2N$ free variables, which is usually ill-posed. One effective way to attack this problem is to introduce regularization, which preserves the regularity of a deformable surface. The following object function is widely used in deformable surface fitting [18, 20, 33, 34] for energy minimization:

$$E(\mathbf{s}) = E_e(\mathbf{s}) + \lambda_r E_r(\mathbf{s}) \tag{3.2}$$

where $E_e(\mathbf{s})$ is the sum of the weighted square error residuals for the matched points. Also, $E_r(\mathbf{s})$ is the regularization term that represents the surface deformation energy, and λ_r is a regularization coefficient.

A set of correspondences M between the model and the input image can be built through a point matching algorithm. Therefore, a pair of matched points is represented in the form of $\mathbf{m} = \{\mathbf{m}_0, \mathbf{m}_1\} \in M$, where $\mathbf{m}_0$ is defined as the 2D coordinates of a feature point in the training image and $\mathbf{m}_1$ is the coordinates of its match in the input image. Then, the correspondence error term $E_e(\mathbf{s})$ is formulated as below:

$$E_e(\mathbf{s}) = \sum_{\mathbf{m} \in M} \omega_{\mathbf{m}} \mathcal{V}(\delta, \sigma) \tag{3.3}$$

where $\mathcal{V}(\delta, \sigma)$ is a robust estimator, and $\omega_{\mathbf{m}} \in [0, 1]$ is a weight linked with each correspondence.

The regularization term E_r in Equ. 3.2, also known as internal force in Snakes [20], is composed of the sum of the squared second-order derivatives of the mesh vertex coordinates. As the mesh is regular, $E_r(\mathbf{s})$ can be formulated through a finite difference:

$$E_r(\mathbf{s}) = \mathbf{s}^\top \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \mathbf{s} \tag{3.4}$$

where K is a sparse and banded matrix which is determined by the structure of the explicit mesh model [13].

3.4.3 Finite Newton Formulation

In this chapter, we employ a robust estimator $\mathcal{V}(\delta, \sigma)$ with compact support size σ. Moreover, δ is the residual error, which is defined as follows:

$$\delta = \mathbf{m}_1 - T_{\mathbf{s}}(\mathbf{m}_0) \tag{3.5}$$

The robust estimator function $\mathcal{V}(\delta, \sigma)$ that assesses a fixed penalty for residuals larger than a threshold σ is employed in the present work; this approach is relatively insensitive to outliers [5]:

$$\mathcal{V}(\delta, \sigma) = \begin{cases} \frac{\|\delta\|}{\sigma^n}, & M_1 = \{\mathbf{m}| \quad \|\delta\| \leq \sigma^2\} \\ \sigma^{2-n}, & M_2 = \overline{M_1} \end{cases} \tag{3.6}$$

where the set M_1 contains the inlier matches, and M_2 is the set of the outliers. In addition, the order n determines the scale of the residual. As shown in Fig. 3.2, the most correspondences are included when the support σ is large. As σ decreases, the robust estimator becomes narrower and more selective.

Since the robust estimator function is not convex, the associated penalty function approximation problem becomes a hard combinational optimization problem. We tackle this problem under the finite Newton optimization framework. An augmented vector $\mathbf{t} \in R^N$ containing the barycentric coordinates is defined as below:

$$\mathbf{t}_i = \xi_1 \quad \mathbf{t}_j = \xi_2 \quad \mathbf{t}_k = \xi_3$$

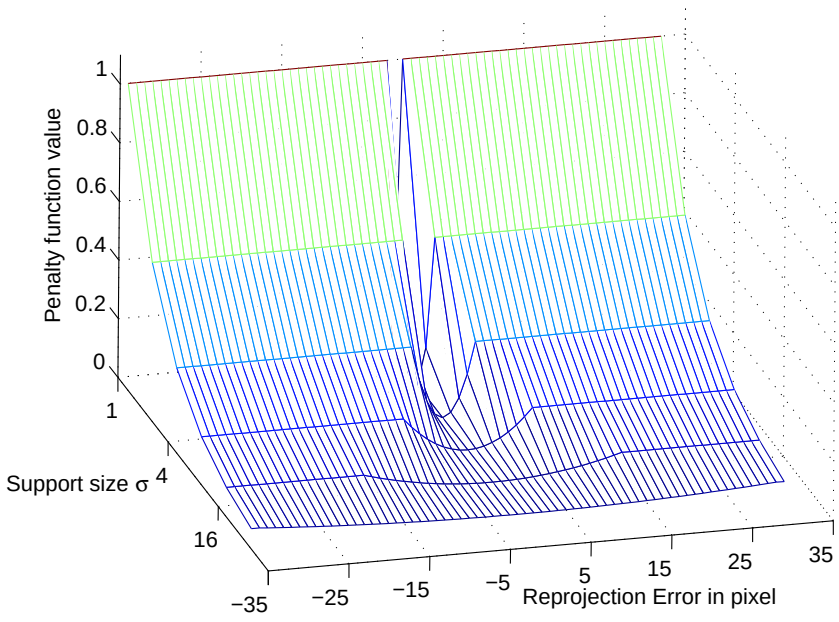


Figure 3.2. The robust estimator that assesses a fixed penalty to residuals larger than a threshold σ.

while the remaining elements in the vector $\mathbf{t}$ are all set to zero. Therefore, the residuals for the inlier correspondences can be rewritten as follows:

$$\begin{aligned} \|\boldsymbol{\delta}\| &= (u - \mathbf{t}^\top \mathbf{x})^2 + (v - \mathbf{t}^\top \mathbf{y})^2 \\ &= u^2 + v^2 - 2(u\mathbf{t}^\top \mathbf{x} + v\mathbf{t}^\top \mathbf{y}) + \mathbf{x}^\top \mathbf{t}\mathbf{t}^\top \mathbf{x} + \mathbf{y}^\top \mathbf{t}\mathbf{t}^\top \mathbf{y} \end{aligned}$$

where (u, v) are the coordinates of $\mathbf{m}_1$. Therefore, the error term in Equ. 3.3 turns out to be

$$E_e = \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n} \left(u^2 + v^2 - 2 \begin{bmatrix} u\mathbf{t} \\ v\mathbf{t} \end{bmatrix}^\top \mathbf{s} + \mathbf{s}^\top \begin{bmatrix} \mathbf{t}\mathbf{t}^\top & 0 \\ 0 & \mathbf{t}\mathbf{t}^\top \end{bmatrix} \mathbf{s} \right) + q\sigma^{2-n}$$

where q is the number of outliers.

Let $\mathbf{b} \in R^{2N}$ be defined as below:

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \end{bmatrix} = \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n} \begin{bmatrix} u\mathbf{t} \\ v\mathbf{t} \end{bmatrix} \tag{3.7}$$

and a matrix $A \in R^{N \times N}$ is equal to

$$A = \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n} \mathbf{t}\mathbf{t}^\top \tag{3.8}$$

Thus, the energy function Equ. 3.3 is formulated into an unconstrained quadratic optimization problem, which can be solved by the modified finite Newton method [26, 22].

$$E = \mathbf{s}^\top \begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathbf{s} - 2\mathbf{b}^\top \mathbf{s} + q\sigma^{2-n} + \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n}(u^2 + v^2)$$

The finite gradient of the energy function E with respect to $\mathbf{s}$ can be derived as below:

$$\nabla = 2\left(\begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \end{bmatrix} \right) \tag{3.9}$$

and the Hessian [5] can also be computed by

$$H = 2 \begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \tag{3.10}$$

Thus the gradient can be rewritten as below:

$$\nabla = H\mathbf{s} - 2\mathbf{b}$$

Each Newton step will perform the following operation:

$$\mathbf{s} \leftarrow \mathbf{s} - \gamma H^{-1} \nabla \tag{3.11}$$

where γ is the step size. Substitute the Equ. 3.9 into Equ. 3.11, we can obtain the equation as follows:

$$\mathbf{s} \leftarrow (1 - \gamma)\mathbf{s} + 2\gamma H^{-1}\mathbf{b}$$

Thus, the update equation can be computed by

$$\mathbf{x} \leftarrow (1 - \gamma)\mathbf{x} + \gamma[\lambda_r K + A]^{-1}\mathbf{b}_x$$

$$\mathbf{y} \leftarrow (1 - \gamma)\mathbf{y} + \gamma[\lambda_r K + A]^{-1}\mathbf{b}_y$$

In our experiment, we simply set γ equal to one, and no convergence problem occurs in our experiments. Therefore, the new update equation can be derived as below:

$$H\mathbf{s} = \mathbf{b} \tag{3.12}$$

Substitute the Equ 3.10 into Equ 3.12, we find that update of the state vector $\mathbf{s}$ can be computed by the following linear equation:

$$\begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathbf{s} = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \end{bmatrix}$$

Since K is regular, the problem can be further simplified into two linear equations which can be efficiently solved via LU decomposition:

$$\mathbf{x} = (\lambda_r K + A)^{-1}\mathbf{b}_x \tag{3.13}$$

$$\mathbf{y} = (\lambda_r K + A)^{-1}\mathbf{b}_y \tag{3.14}$$

The overall complexity is thus the complexity of one Newton step. Moreover, the complexity of one step for the proposed method is same as [19].

3.4.4 Progressive Finite Newton Optimization

Generally speaking, the incorrect matches cannot be avoided in the first stage of the matching process where only local image descriptors are compared. We introduce a coarse-to-fine scheme to deal with those outliers. The support σ of robust estimator $\mathcal{V}(\delta, \sigma)$ is progressively decayed at a constant rate α. Since the derivatives of $\mathcal{V}(\delta, \sigma)$ are inversely proportional to the support σ, the regularization coefficient λ_r is kept constant during the optimization. For each value of σ, the object function E is minimized through the finite Newton step and the result is employed as the initial state for the next minimization. The minimization of E is directly solved through Equ. 3.13 and Equ. 3.14 for a given initial state, and one step is enough to achieve convergence. The optimization procedure stops when σ reaches a value close to the expected precision, which is usually one or two pixels. The algorithm reports a successful detection when the number of inlier matches is above a given threshold. Thus, the whole optimization problem can be solved within a fixed number of steps. This is in contrast to the semi-implicit optimization scheme [34], which involves a few iterations for each σ, and at least 40 iterations in total to ensure the convergence.

In order to select most of the correspondences into the initial active set and avoid getting stuck at local minima, the initial value of σ is usually set to a sufficiently large value. However, this requires a fixed initial state. The method is dependent on the object position, and needs a few iterations to compensate for the errors generated by the pose variations. In the present work, we solve this problem through a modified RANSAC approach. Taking advantage of our concise finite Newton formulation and closed-form solution, the explicit mesh can be directly estimated from a given set of correspondences. Moreover, we draw from progressively larger sets of top-ranked correspondences, which decreases the number of trials significantly. In the experiments, the sampling process stopped within 5 trials. In the worst case, such as when an object does not appear in the scene, it still converges towards RANSAC. Therefore, the output of the proposed progressive sample can be employed as the initial state for the finite Newton optimization. Since the result of progressive sample estimation is quite close to the solution,

(a) Model image (b) Result (c) Result (d) Result (e) Plastic cup

Figure 3.3. We use a Starbucks pad as the deformable object. The model image is shown in (a) the contour of the model image is extracted using a simple gradient and filling operator, which is overlaid on the input image. (b) to (d) show the results. (e) shows the result where the pad is replaced by a plastic cup. The model contains 120 vertices, and the whole process, including image capturing and rendering, runs around 18 frames per second. More results are included in the supplementary material.

σ is relatively small. Thus, the proposed progressive scheme requires fewer stages, and is somewhat invariant to the initial position.

3.5 Experimental Results

In this section, we discuss the details of our experimental implementation and report the results of performance evaluation on nonrigid surface detection. We show that the proposed approach is very efficient for real-time tracking, and can be easily employed for augmented reality applications. In addition, same convincing results are obtained for medical image registration, even with missing data.

3.5.1 Experimental Setup

In order to register the mesh model conveniently, a model image is acquired when the nonrigid surface contains no deformation. In order to facilitate real-time augmented reality applications, a random-trees based method [23] is used to build the correspondences between the model image and input image.

Since the number of free variables for nonrigid surface recovery is usually quite large (even up to one thousand),

the sample size of each RANSAC iteration becomes a tricky issue. We compare the performance with different sample sizes. In our experiments, the support σ is empirically set to 30, and λ_r is set to a large value to ensure the regularity of the nonrigid surface. Interestingly, we find that the best sample size is three. This is because the nonrigid surface degenerates into a rigid one, and only three points are necessary to determine the position of a rigid surface. Moreover, when the sample size increases, the probability of selecting the inlier data is decreased. Thus, three is the best choice for the sample size.

In the finite Newton optimization, the weighting scheme is beneficial for a single step. However, it changes the scale of the error term in the object function, and so the regularization coefficient λ_r is no longer kept constant during the optimization. In our experiments, all weight coefficients ω are set to one. A set of synthetic data is used to select the parameters, and the reference mesh is manually registered. The performance is evaluated by the percentage of mesh vertices within 2 pixels of those in the reference mesh. The best regularization coefficient is found to be around 3×10^{-4} by grid searching. Similarly, the initial support σ_0 is set to 80, and decay rate α is 0.5. Fig. 3.4 plots the success probability with different orders n of the robust estimator function. Based on these results, n is set to 4.

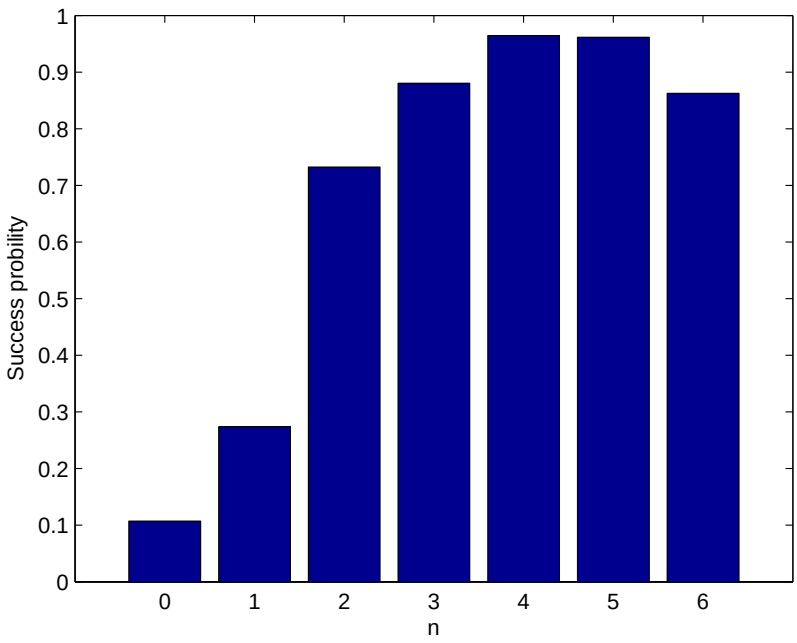


Figure 3.4. Probability of success with different order n of the robust estimator function.

All the experiments reported in this chapter were carried out on a Pentium-4 3.0GHz PC with 1GB RAM, and

a DV camera was engaged to capture video. We also implemented a semi-implicit iterative method [34], which is regarded as the state-of-the-art approach.

3.5.2 Computational Efficiency

The complexity of the proposed method is mainly dominated by the order of Equ. 3.13 and Equ. 3.14, which is equal to the number of vertices N in the mesh model. Another important factor is the number of inlier matches, which affects the sparseness of matrix A. This usually differs from one frame to another. For the Starbucks pad with 120 vertices, as shown in Fig. 3.3, the proposed method runs at 18 frames per second on real-time video with size of 720×576. As depicted in Table 3.1, the proposed optimization scheme requires around 8 iterations and only takes half of the time of the feature matching algorithm, which is the bottleneck of the whole system. Our implementation [1] of semi-implicit iterative approach [34] needs around 40 iterations to reach the convergence, and runs about 9 frames per second. The improvement is more significant for high resolution mesh. Thus, the proposed method requires far less iterations, and is efficient for real-time applications. We also conduct the experiments without using the modified RANSAC initialization, and start the optimization scheme from a sufficiently large support $\sigma = 1000$. This requires 11 iterations, and the fitting accuracy is worse than the proposed method. In addition, the modified RANSAC initialization can also be used for a semi-implicit method, in which case the number of iterations is reduced to around 25.

Table 3.1. Computational time of proposed method at each step.

Total	Match	Optimization	Iteration	Other
57ms	27ms	14ms	$\sim$ 1.9ms	16ms

3.5.3 Performance of Nonrigid Surface Recovery

We use a Starbucks pad as the deformable object. As shown in Fig. 3.3, the proposed method is robust to large deformations and perspective distortion. In practice, the whole process runs at around 18 frames per second. Fig. 3.6 describes the result of detecting a piece of paper, where similar performance is achieved. As another feature-based method, the performance of the proposed method is closely related to the texture of objects. Better results can be obtained for objects with more texture, because it is easy to find more correct correspondences than

[1] We use the same parameters setting as [34]. The convergence condition is set to 0.9995, with at most 5 iterations for each support value σ.

with those lacking texture. This problem may be solved by incorporating global appearance and edge constraints into the optimization scheme.

Figure 3.5. Re-texturing of a shirt print. The first row and third row show the 720×576 images captured by a DV camera. The second row and forth row show the results of replacing the bunny with the CVPR logo.

3.5.4 Augmented Reality

Once the nonrigid surface is recovered, an immediate application is to re-texture an image. In order to obtain realistic results, the texture should be correctly relighted. As suggested in [34], a re-textured input image is

Figure 3.6. Re-texturing a picture on a piece of paper. The first row is the 720×576 images captured by a DV camera. The second row is the results of replacing the picture with the CVPR logo.

generated by directly multiplying a blank shaded image, which is the quotient of the input image and the warped reference image. The reference image is acquired when the nonrigid surface is lighted uniformly. Moreover, the quotient image is normalized through multiplying the intensity of white color in the reference image. This relighting procedure is easily done by the GPU and requires only a short OpenGL shading language program; and the whole process runs at about 17 frames per second. Fig. 3.5 shows the results of re-texturing a T-shirt with a Lambertian surface. It is difficult to estimate a blank shaded image due to dividing near zero intensity values and the use of an uncontrolled optical sensor. However, the visual effect is that the bunny in the input video is re-textured by the CVPR logo. For a specular surface, Fig. 3.6 describes the results on a piece of paper with a saturated region. In addition, the right two columns of Fig. 3.6 show the results in a cluttered environment.

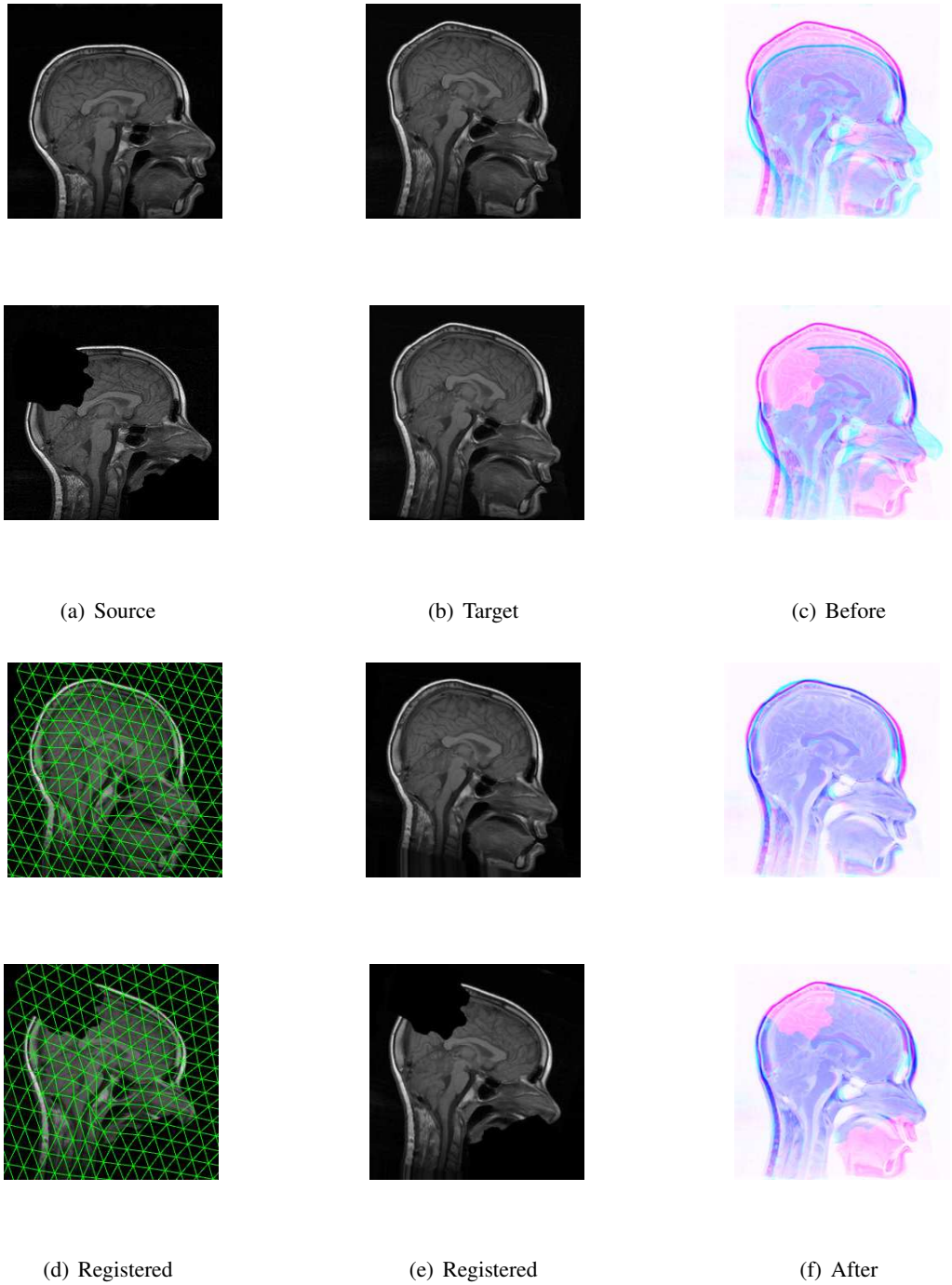
3.5.5 Medical Image

We also evaluate the proposed approach for medical image registration. A pair of sagittal images [31] with size of 256×256 from two different patients are used in the experiments. The source and target images differ in both geometry and intensity. The results are plotted in Fig. 3.7; it can be seen that the source image is successfully registered. In comparison with the locally affine but globally smooth method [31], which takes about 4 minutes, our proposed method only needs 0.2 seconds. Moreover, the sparse correspondences based method can naturally handle the missing data and partial occlusion problem. As shown in the second row of Fig. 3.7, the source images with a region removed, the nonrigid shape can still be recovered.

3.6 Discussions and Future Work

We have proposed a novel scheme for non-rigid surface detection by progressive finite Newton optimization. In comparison with semi-implicit optimization methods [34], the proposed method has several advantages. First, we need not solve the optimization iteratively for every σ, because it can be solved in one step directly. Second, the iterative method starts from a sufficiently large support value in order to estimate the location and pose of an object, which leads to a large number of iterations. Thus, the proposed method is far more efficient than the semi-implicit method. Additionally, it is easy to implement the proposed approach, which only involves solving the sparse linear equation, and does not require tuning the viscosity parameters and a sophisticated Levenberg-Marquardt optimization algorithm.

Although promising experimental results have validated the efficiency of our methodology, some limitations and future directions should be addressed. First of all, some jitter may occur due to the point matching algorithm or the lack of texture information. Also, we have focused our attention only on single deformable surface detection,



(a) Source (b) Target (c) Before

(d) Registered (e) Registered (f) After

Figure 3.7. Applying the proposed method to medical image registration. A pair of sagittal images from two different patients is shown. (a,b,e) are the source, target and registered source respectively. (d) is the registered source with mesh model. (c) and (f) are the overlaid images before and after registration. The second row displays the synthetic example with missing data.

whereas it is also interesting to study the multiple case. In future work, global bundle-adjustment will be introduced to tackle the jittering problem. Furthermore, an efficient octree structure can be used to build a simplified multi-scale mesh model. Finally, we may consider extending the proposed scheme to 3D environments and exploring new regularization methods.

3.7 Summary

This chapter presented a novel progressive scheme to solve the non-rigid surface detection problem. In contrast to the previous approaches involving iterative and explicit minimization, we proposed a progressive finite Newton algorithm, which directly solves the unconstrained quadratic optimization problem by an efficient factorization method. Moreover, our modified RANSAC scheme takes advantage of our concise formulation and progressive sampling of the top-ranked correspondences, and can handle high-dimensional spaces with noisy data.

We have conducted extensive experimental evaluations on diverse objects with different materials. The proposed method is very fast and robust, and can handle large deformations and illumination changes. It was tested in several applications, such as real-time Augmented Reality and medical image registration. The promising experimental results showed that our algorithm is more efficient and effective than previous methods.

Chapter 4

Conclusion

In this report, two different approaches have been proposed to tackle the nonrigid surface modelling and recovery problem. First, an appearance-based approach with a two-stage scheme is presented for online 3D facial features and pose estimation, which employs the PCA based regularization method. Second, a fast and automated nonrigid surface recovery method is proposed, which takes advantage of the progressive Newton optimization method. Moreover, this approach is based on the feature correspondences; and the surface deformation is constrained by a physical model.

In contrast to the feature-based methods, the appearance-based approach may tend to be computational expensive and require good initialization. However, it will be probably more accurate than the feature-based methods, and has the representation capability. On the other hand, compared with the appearance-based methods, the feature based approach is fast, which is able to take advantage of the recent advance in object recognition. Moreover, the efficient and automated solution has already been proposed in literature. In future, the fusion approach will be studied, which may take the advantage of both the global appearance information and local features.

Acknowledgments

I would like to thank Prof. Michael R. Lyu for his kind guidance. I also thank Dr. Steven C.H. Hoi for his help. Moreover, I thank Dr. Haixuan Yang for providing the template of this report.

Bibliography

[1] J. Ahlberg. Using the active appearance algorithm for face and facial feature tracking. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pages 68–72, 2001.

[2] A. Bartoli and A. Zisserman. Direct estimation of non-rigid registration. In *Proc. British Machine Vision Conference*, Kingston, Sep. 2004.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[4] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9), 2003.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] C.Bregler, A.Hertzmann, and H.Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 690–696, 2000.

[7] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

[8] O. Chum and J. Matas. Matching with prosac- progressive sample consensus. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 220–226, 2005.

[9] T. Cootes, G. Edwards, and C. Taylo. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6), June 2001.

[10] T. Cootes and P. Kittipanya-ngam. Comparing variations on the active appearance model algorithm. In *Proc. British Machine Vision Conference*, volume 2, pages 837–846, 2002.

[11] D. DeCarlo and D. N. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Int'l J. Computer Vision*, 38(2):99–127, 2000.

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981.

[13] P. Fua and Y. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *Int'l J. Computer Vision*, 16(1):35–56, Sep. 1995.

[14] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. In *British Machine Vision Conference*, September 2004.

[15] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *SIGGRAPH'1998*, pages 477–486, 1998.

[16] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[17] X. Hou, L. S.Z., H. Zhang, and Q. Cheng. Direct appearance models. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 828–833, 2001.

[18] S. Ilic and P. Fua. Implicit meshes for surface reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(2):328–333, February 2006.

[19] S. Ilic, M. Salzmann, and P. Fua. Implicit meshes for effective silhouette handling. *Int'l J. Computer Vision*, 2006.

[20] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int'l J. Computer Vision*, 1(4):321–331, Jan. 1988.

[21] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, Oct. 2004.

[22] S. S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research*, 6:341–361, 2005.

[23] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2006.

[24] H. Ling and D. W. Jacobs. Deformation invariant image matching. In *Proc. Int'l Conf. Computer Vision*, pages 1466–1473, 2005.

[25] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int'l J. Computer Vision*, 60(2):91–110, 2004.

[26] O. L. Mangasarian. A finite newton method for classification. *Optimization Methods and Software*, 17(5):913–929, 2002.

[27] G. Medioni and S. B. Kang. *Emerging topics in computer vision*. Prentice Hall, 2004.

[28] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[29] P. Mittrapiyanumic, G. DeSouza, and A. Kak. Calculating the 3d-pose of rigid-objects using active appearance models. In *Proceedings of the International Conference in Robotics and Automation*, volume 5, pages 5147–5152, 2004.

[30] G. Nielson. Scattered data modeling. *IEEE Computer Graphics and Applications*, 13, 1993.

[31] S. Periaswamy and H. Farid. Medical image registration with partial data. *Medical Image Analysis*, (10):452–464, 2006.

[32] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH'1998*, pages 75–84, 1998.

[33] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 822–828, 2005.

[34] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration, and realistic augmentation. *Int'l J. Computer Vision*, 2007.

[35] M. Stegmann, B. Ersboll, and R. Larsen. Fame-a flexible appearance modeling environment. *IEEE Trans. on Medical Imaging*, 22(9), 2003.

[36] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *IEEE Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 493–500, 2001.

[37] L. V. Tsap, D. B. Goldgof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(5):526–543, 2000.

[38] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6), Oct. 2004.

[39] R. White and D. Forsyth. Retexturing single views using texture and shading. In *Proc. European Conf. Computer Vision*, pages 70–81, 2006.

[40] R. White and D. A. Forsyth. Combining cues: Shape from shading and texture. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1809–1816, 2006.

[41] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 535–542, 2004.

[42] J. Zhu, S. C. Hoi, and M. R. Lyu. Real-time non-rigid shape recovery via active appearance models for augmented reality. In *Proc. European Conf. Computer Vision*, pages 186–197, 2006.

[43] J. Zhu, S. C. Hoi, and M. R. Lyu. A multi-scale tikhonov regularization scheme for implicit surface modelling. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007, To appear.

[44] J. Zhu, S. C. Hoi, and M. R. Lyu. Face annotation by transductive kernel fisher discriminant. *IEEE Trans. on Multimeida*, 2007, under major revision.

[45] J. Zhu, S. C. Hoi, E. Yau, and M. R. Lyu. Automatic 3d face modeling using 2d active appearance models. In *Proc. 13th Pacific Conf. Computer Graphics and Applications*, pages 133–135, 2005.

[46] J. Zhu and M. R. Lyu. Progressive finite newton approach to real-time nonrigid surface detection. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007, To appear.